Minimum cost flow problem

CE 367R
MINIMUM COST FLOW PROBLEM
Respecting capacities, find link flows which balance supply and demand among sources and sinks, with minimum total cost.
Applications

- Logistics and shipping
- Earthwork in roadway construction
- Passenger selection in ridesharing
- Labor assignments in a co-op
Each link \((i, j)\) has a specified cost \(c_{ij}\) and capacity \(u_{ij}\), and we must determine its flow \(x_{ij}\).

Each node has a supply \(b_i\). If \(b_i > 0\), the node is a source, if \(b_i < 0\) it is a sink, and if \(b_i = 0\) it is a transhipment node.

We want to minimize the total transportation cost \(\sum_{(i,j)\in A} c_{ij}x_{ij}\).

Constraints are the link capacities, and flow conservation.

For simplicity, and WLOG, assume that if there is a link \((i, j)\), then there is no link in the reverse direction \((j, i)\).
Minimum cost flow problem

$$\min_x \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.t. $$\sum_{(i,j) \in A(i)} x_{ij} - \sum_{(h,i) \in B(i)} x_{hi} = b_i \quad \forall i \in N$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$
Example
Supply chain logistics can often be represented by a min cost flow problem.

The following model is based on Shahabi, Unnikrishnan, Shirazi & Boyles (2014). A three-level location-inventory problem with correlated demand. *Transportation Research Part B* 69, 1–18.
There are three kinds of nodes, representing manufacturing plants, warehouses, and retailers. Each manufacturing plant produces a certain quantity of product, which must be shipped first to a warehouse, and then from a warehouse to a retailer which will sell a prespecified quantity of product.

(The paper considered other factors: locations of plants and warehouses were decision variables, and demand was a random variable not known in advance.)
There is a transportation cost between each plant and warehouse, and between each warehouse and retailer. These transportation links have a capacity; furthermore, each warehouse has a capacity.
Earthwork in road construction

A road must be built through mountainous terrain; how can we transport soil between areas of cut and fill for the least cost? Soil can be brought on-site or taken off-site for a higher cost.

Node supply/demand indicates amount of cut or fill; links indicate on-site transport with costs per unit of soil. Additional links (not seen) can connect to an artificial “off-site” node.
Imagine a fixed-route jitney operation that accepts reservations a day in advance for a vehicle with a given number of seats. The driver can choose which requests to serve to maximize profit.

Create a node for each stop, and a link connecting each stop with capacity equal to vehicle seats, and cost equal to the negative of fare per passenger. Create artificial links between each pair of stops with zero cost and infinite capacity for unserved demand.
Labor assignments in a co-op

Cooperative housing offers students cheaper rent in exchange for performing labor (cooking, cleaning, etc.).

In a large co-op, it is difficult to balance the needs of the house with the preferences and availability of residents.

Enter ATLAS (Automated Taos Labor Assignment Scheduling)
Two sets of nodes are created: one per resident, and one per task. The “supply” for each resident is the number of hours they must work, the “demand” for each task node is the number of hours needed.

Links are created between each resident and task node, with a cost related to how unpleasant it would be for that person to do that task (can be $\infty$ if it is impossible due to scheduling, etc.)
ATLAS reduced the number of scheduling conflicts and was successfully received.

It’s so awesome!!!!!!!!!!!!!!!!!!
Rj, this things pretty badass.
It is a fine structure, this site which lay before mine eyes as summer grass before the yawning sun. As a high priest of Taos I bless it, and bless you, most elegant sir.
It could stand to be much prettier.
RELATION TO OTHER PROBLEMS
The minimum cost flow problem can be seen as a generalization of the shortest path and maximum flow problems.

That is, by suitably choosing costs, capacities, and supplies we can solve shortest path or maximum flow using any method which will solve min cost flow.

(Naturally, this means that solving the minimum cost flow problem must be at least as hard as solving shortest path or max flow.)
Transformation to shortest path problem

For each link, set $u_{ij}$ to $\infty$. (Capacities ignored in shortest path.)
For the origin $s$, set $b_s = +1$. For the destination $t$, set $b_t = -1$. For all other nodes $i$, $b_i = 0$. 
Transformation to maximum flow problem

Create an artificial link \((t, s)\) connecting the sink \(t\) to the source \(s\), with capacity \(u_{ts} = \infty\).

Set \(c_{ij} = 0\) for all links, except the artificial link, which gets cost \(c_{ts} = -1\).
NEGATIVE CYCLE CANCELLING ALGORITHM
Like the augmenting path algorithm, the negative cycle cancelling algorithm is based on finding an undirected path — in this case an undirected cycle.

As before, the residual capacity of a forward link is $u_{ij} - x_{ij}$, and the residual capacity of a reverse link is $x_{ij}$.

The cost of a forward link is $c_{ij}$, the cost of a reverse link is $-c_{ij}$.

Forward links show what happens if we increase the flow on a link, reverse links show what happens if we decrease the flow on a link.
Algorithm

1. Generate any feasible flow \( x \).
2. Attempt to find an undirected cycle which has positive residual capacity and negative cost (call such a cycle \( C \)). If no such cycle exists, terminate: \( x \) is an optimal flow.
3. Let \( \bar{u}_C \) be the minimum residual capacity of the links in \( C \).
4. Update \( x \) by adding \( \bar{u}_C \) to the flow on all forward links in \( C \), and by subtracting \( \bar{u}_C \) from all reverse links in \( C \).
5. Return to step 2.
Minimum cost flow problem

Negative cycle cancelling algorithm
Minimum cost flow problem

Negative cycle cancelling algorithm
Minimum cost flow problem

Negative cycle cancelling algorithm
Minimum cost flow problem

Negative cycle cancelling algorithm
FILLING IN THE "HANDWAVING"
How to find an initial flow?

Ignore costs, solve max flow on the transformed network!
How to find a negative cost cycle?

We can do this “by inspection” — which is enough for homework problems — but this is a big hand-wave.

Fact: if the network is strongly connected and there is a negative cost cycle, the shortest path problem does not have a solution. (Why?)

If $C$ is the largest cost of any link, and the label-correcting algorithm ever assigns an $L_i$ value less than $-nC$, there must be a negative cycle, and we can trace it back from the backnodes starting at $i$. 

Minimum cost flow problem

Filling in the “handwaving”
To find undirected cycles with positive residual capacity, modify the label correcting algorithm in the following way:

- When scanning a node $i$, instead of just looking at links $(i, j)$ in its forward star, also look at links $(h, i)$ in the reverse star (these would correspond to reverse links in the path).
- Skip any forward links whose flow is at capacity, and any reverse links whose flow is zero.
- For forward links, scan as before. For reverse links, compare $L_h$ to $L_i - c_{hi}$, and if the latter is smaller, set $L_h = L_i - c_{hi}$ and $q_h = i$. 

Minimum cost flow problem

Filling in the “handwaving”
SUCCESSIVE SHORTEST PATH ALGORITHM
Finding negative cycles with label-correcting can take a long time. An alternative method is the *successive shortest path algorithm*.

For each node $i$, let $\pi_i$ be its *potential*. (You can think of this as an energy potential from physics or head from fluids; this will play a similar role to the $L$ values in shortest path.)

For each link $(i, j)$ let its reduced cost be $\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j$, and let $\bar{c}_{ji} = -\bar{c}_{ij}$.

We can show that $\mathbf{x}$ is optimal iff there exist potentials $\pi$ such that $\bar{c}_{ij} \geq 0$ for every link for which $x_{ij} < u_{ij}$, and $\bar{c}_{ij} \leq 0$ for every link for which $x_{ij} > 0$. 

*Minimum cost flow problem*  
*Successive shortest path algorithm*
The successive shortest path algorithm builds up a flow from zero. The imbalance of a node is given by

\[ e(i) = b(i) + \sum_{(h,i) \in B(i)} x_{hi} - \sum_{(i,j) \in A(i)} x_{ij} \]

A node with zero imbalance satisfies flow conservation; if \( e(i) > 0 \) there is a surplus at node \( i \), if \( e(i) < 0 \) there is a deficit.
Successive shortest path algorithm

1. Let $x = 0$ and $\pi = 0$.
2. Let $e(i) = b_i$ for all nodes $i$.
3. If $e(i) = 0$ for all $i$, stop. Otherwise, select a node $s$ for which $e(s) > 0$ and a node $t$ for which $e(t) < 0$.
4. Find undirected shortest paths with residual capacity from $s$ to all other nodes, with respect to the reduced costs $\tilde{c}$. Let $L$ be the shortest path labels, and $P$ the shortest path from $s$ to $t$.
5. Add $L$ to $\pi$.
6. Let $\delta$ be the residual capacity of $P$ (minimum of $u_{ij} - x_{ij}$ for forward arcs, and $x_{ij}$ for reverse arcs).
7. Increase the flow on the forward arcs of $P$ by $\delta$, and decrease flow on reverse arcs by $\delta$. Subtract $\delta$ from $e(s)$, and add $\delta$ to $e(t)$.
8. Return to step 3.
Example

Minimum cost flow problem

Successive shortest path algorithm
PRIMAL-DUAL ALGORITHM
The primal-dual algorithm is an enhancement of successive shortest paths, which sends flow along all shortest paths at the same time, rather than just one.

We assume the network has a single source, and a single sink. As shown on the next slide, this imposes no loss of generality.
We also transform the network to have a single source node $s$ and a single sink node $t$ with the following procedure:

Create the *supersource* and *supersink* nodes $s$ and $t$.

Create links between the supersource $s$ and every source $i$, with zero cost and capacity $b(i)$. Change $b(i)$ to zero, and make $b(s)$ equal to the total supply in the network.

Do similarly with links between every sink and the supersink.

It is trivial to convert a flow in one network to a flow in the other, with equal cost.
Primal-dual algorithm

1. Let $x = 0$ and $\pi = 0$.
2. Let $e(s) = b(s)$ and $e(t) = b(t)$.
3. If $e(s) = 0$, stop. Otherwise, continue.
4. Find shortest path labels $L$ from origin $s$ to all other nodes, with respect to the reduced costs $\bar{c}$.
5. Add $L$ to $\pi$.
6. Create a new network $G'$ which contains links $(i, j)$ only if $\bar{c}_{ij} = 0$ or $\bar{c}_{ji} = 0$; set the capacity of each link in $G'$ equal to the residual capacity of the corresponding link in $G$.
7. Solve the maximum flow problem from $s$ to $t$ in $G'$.
8. Augment flow from this solution, updating $e(s)$, $e(t)$, and $x$ (add flow for forward links, subtract for reverse).
Example

Minimum cost flow problem

Primal-dual algorithm